Global Workforce Development Initiatives: *GSwE2009* and *BKCASE*

Presented at DISA

4 January 2010

Art Pyster

art.pyster@stevens.edu

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **04 JAN 2010**	2. REPORT TYPE	3. DATES COVERED **00-00-2010 to 00-00-2010**
4. TITLE AND SUBTITLE **Global Workforce Development Initiatives: GSwE2009 and BKCASE**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Stevens Institute of Technology,Body of Knowledge and Curriculum to Advance Systems Engineering (BKCASE),1 Castle Point on Hudson,Hoboken,NJ,07030**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES **BKCASE is sponsored by the Department of Defense.**

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**30**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

What is GSwE2009?

- GSwE2009 is a set of recommendations for faculty who are creating or updating a graduate program in software engineering (SwE)

- Secondarily, it could be used by
  - employers in selecting new SwE graduates, and
  - students in selecting graduate programs

- GSwE2009 is intended for world-wide use

- GSwE2009 is *not* intended to be used directly for accreditation

- Two companion documents released last November

GSwE2009 Structure

- ***Guidance for Constructing and Maintaining GSwE2009:*** the fundamental principles, assumptions, and context for the GSwE2009 authors

- ***Outcomes:*** what students should have achieved when they graduate

- ***Entrance Expectations:*** what students should be capable of and have experienced when they enter a graduate program

- ***Architecture:*** the structure of a curriculum to accommodate core material, university-specific material, and elective material

- ***Core Body of Knowledge (CBOK):*** material that all students should master in a graduate SwE program

GSwE2009 Operational Concept

- For a program that fully satisfies the GSwE2009 recommendations:

  Each student should arrive meeting all entrance expectations, participate in a program that follows the architecture, master the entire CBOK, and achieve all the outcomes

- HOWEVER, because this is *a reference curriculum*, actual programs will likely choose to deviate from some GSwE2009 recommendations – *this is both expected and reasonable*

Motivation for GSwE2009

- The last effort to create a reference curriculum for graduate software engineering education was by the SEI in the early 1990s.

- There are, in effect, no current community-endorsed recommendations on what to teach software engineers – nothing that recognizes how the world has changed

  - Internet, e-commerce, ubiquity of connectedness

  - Distributed development projects and highly distributed software

  - Embedded software drives value of most new products - interdependence of systems engineering and SwE

- Response: create a project to create a new reference curriculum in software engineering

Extremely Diverse Real Programs

SWEBOK Coverage in required and semi-required courses
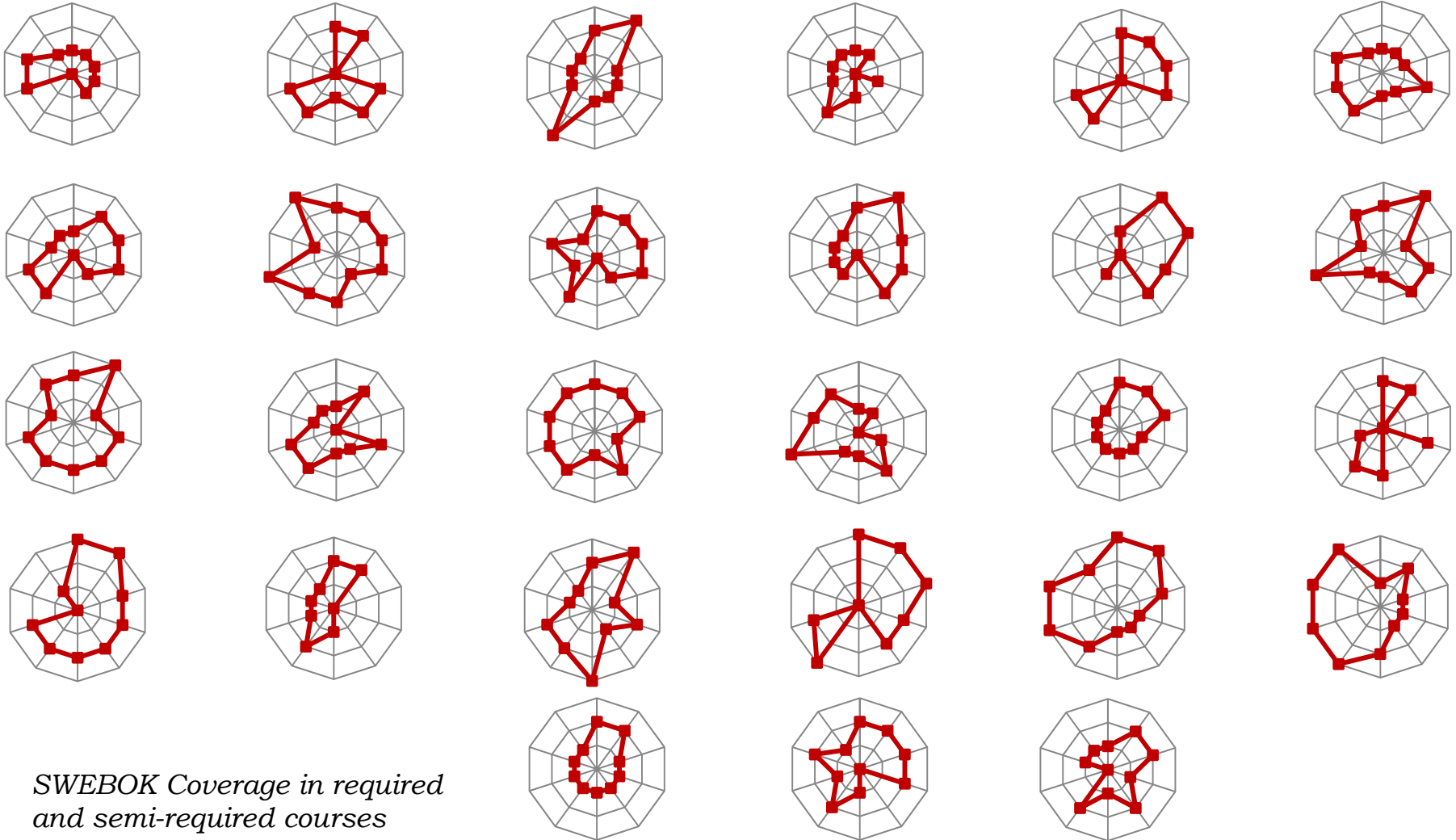
Project History and Process

1. Project to create Graduate Software Engineering Reference Curriculum (GSwERC) started in July 2007 at Stevens Institute of Technology with Department of Defense funding (GSwERC renamed in August to GSwE2009)

2. DoD agreed at beginning of project to take a "hands off" approach to technical content – critical to achieving primary objective

3. Formed Early Start Team of about 15 authors who met in August 2007 to shape project

4. Early Start Team gradually expanded and became Curriculum Author Team (CAT)

5. Workshop held every 3 months to synchronize work, adjust plan, and approve interim products – workshop minutes posted on website

6. Email, WebEx, and teleconferences to conduct business between workshops

7. Teams formed to work on specific sections of GSwE2009

8. Open and transparent operations at all times

Author Team

- Rick Adcock, *Cranfield University and INCOSE*
- Edward Alef, *General Motors*
- Bruce Amato, *Department of Defense*
- Mark Ardis, *Stevens Institute of Technology*
- Larry Bernstein, *Stevens Institute of Technology*
- Barry Boehm, *University of Southern California*
- Pierre Bourque*, École de technologie supérieur*e *and SWEBOK*
- John Bracket, *Boston University*
- Murray Cantor, *IBM*
- Lillian Cassel, *Villanova and ACM*
- Robert Edson, *ANSER*
- Richard Fairley, *Colorado Technical University*
- Dennis Frailey, *Raytheon & Southern Methodist University*
- Gary Hafen, *Lockheed Martin and NDIA SE*
- Thomas Hilburn, *Embry-Riddle Aeronautical University*
- Greg Hislop, *Drexel University and IEEE Computer Society*
- David Klappholz, *Stevens Institute of Technology*
- Philippe Kruchten, *University of British Columbia*
- Phil Laplante, *Pennsylvania State University, Great Valley*
- Qiaoyun (Liz) Li, *Wuhan University, China*

- Scott Lucero, *Department of Defense*
- John McDermid, *University of York, UK*
- James McDonald, *Monmouth University*
- Ernest McDuffie, *National Coordination Office for NITRD*
- Bret Michael, *Naval Postgraduate School*
- William Milam, *Ford*
- Ken Nidiffer, *Software Engineering Institute*
- Art Pyster, *Stevens Institute of Technology*
- Paul Robitaille, *Lockheed Martin and INCOSE*
- Mary Shaw, *Carnegie Mellon University*
- Sarah Sheard, *Third Millenium Systems*
- Robert Suritis*, IBM*
- Richard Thayer*, California State University at Sacramento*
- Barrie Thompson, *Sunderland University, UK*
- Massood Towhidnejad, *Embry-Riddle Aerronautical University*
- Guilherme Travassos, *Brazilian Computer Society, Brazil*
- Richard Turner, *Stevens Institute of Technology*
- Joseph Urban, *Texas Technical University*
- Ricardo Valerdi, *MIT & INCOSE participant*
- Osmo Vikman, *Nokia*
- David Weiss, *Avaya*
- Mary Jane Willshire, *Colorado Technical University*

8

More History and Process

9. Built GSwE2009 on foundation documents, primarily SWEBOK, SE2004, and INCOSE Handbook

10. Created GSwE2009 incrementally and iteratively – Version 0.25 in February 2008, Version 0.5 in October 2008, Version 1.0 in September 2009

11. Invited reviewers for Version 0.25, unlimited review for Version 0.5

12. More than 100 reviewers from 23 countries: Australia, Brazil, Canada, China, Egypt, France, Germany, Greece, India, Israel, Italy, Japan, Latvia, Mexico, Netherlands, Poland, Portugal, Russia, Singapore, Sweden, Thailand, UK, US

13. About 800 review comments for Version 0.5, each adjudicated

14. Numerous presentations and workshops to obtain feedback, and to generate awareness and demand:

   – CSEET 2009, ICSE 2009, SIGCSE 2008 and 2009; NDIA Systems Engineering Conferences 2007, 2008, and 2009; INCOSE International Symposium 2008 and 2009; ASEE 2008; Asian-Pacific INCOSE Conference 2008; …

Expectations at Entry to a Program

DEGREE:

The equivalent of an undergraduate degree in computing, or an undergraduate degree in an engineering or scientific field and a minor in computing

SWE COURSE:

The equivalent of an introductory course in software engineering

EXPERIENCE:

At least two years of practical experience in some aspect of software engineering or software development. This experience should include participation in teams, development of a program or component that has been successfully delivered, and an update or repair to an existing program or component.

Outcomes at Graduation

CBOK:

Master the Core Body of Knowledge

DOMAIN:

Master software engineering in at least one application domain, such as finance, medical, transportation, or telecommunications, and in one application type, such as real-time, embedded, safety-critical, or highly distributed systems. That mastery includes understanding how differences in domain and type manifest themselves in both the software itself and in their engineering, and includes understanding how to learn a new application domain or type.

DEPTH:

Master at least one knowledge area or sub-area from the Core Body of Knowledge to at least the Bloom Synthesis level.

Outcomes at Graduation

ETHICS:

Be able to make ethical professional decisions and practice ethical professional behavior.

SYSTEMS ENGINEERING:

Understand the relationship between software engineering and systems engineering and be able to apply systems engineering principles and practices in the engineering of software.

TEAM:

Be an effective member of a team, including teams that are multinational and geographically distributed, effectively communicate both orally and in writing, and lead in one area of project development, such as project management, requirements analysis, architecture, construction, or quality assurance.

Outcomes at Graduation

RECONCILIATION:

Be able to reconcile conflicting project objectives, finding acceptable compromises within limitations of cost, time, knowledge, risk, existing systems, and organizations.

PERSPECTIVE:

Understand and appreciate feasibility analysis, negotiation, and good communication with stakeholders in a typical software development environment, and perform those tasks well; have effective work habits and be a leader.

LEARNING:

Be able to learn and apply new models, techniques, and technologies as they emerge, and appreciate the necessity of such continuing professional development.
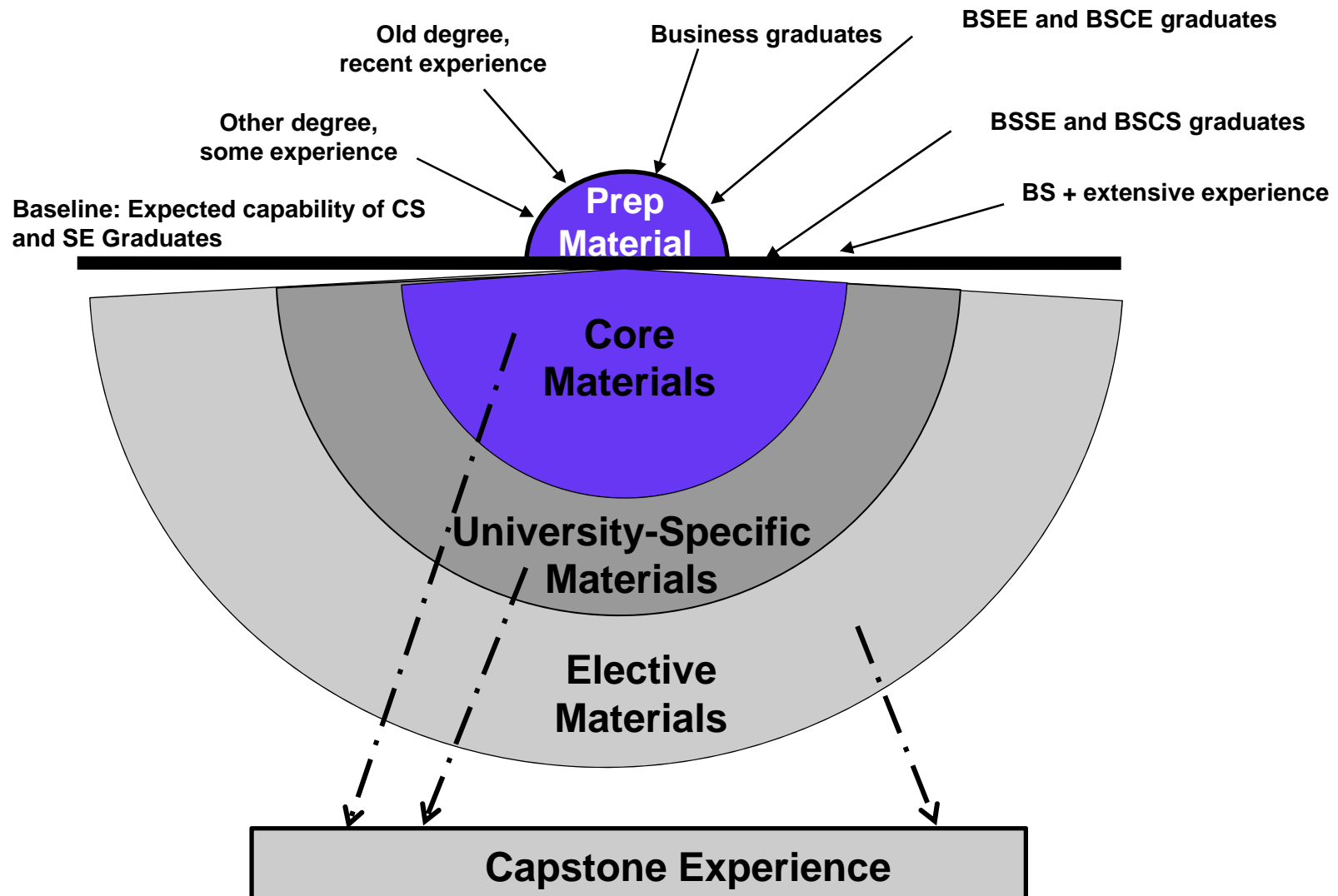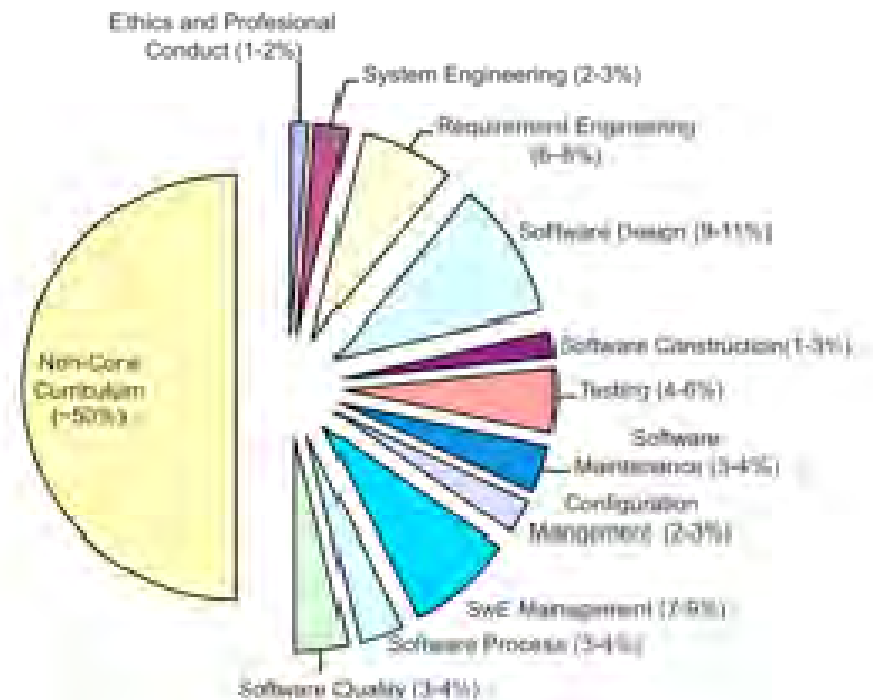
Outcomes at Graduation

TECHNOLOGY:

Be able to analyze a current significant software technology, articulate its strengths and weaknesses, compare it to alternative technologies, and specify and promote improvements or extensions to that technology.

Curriculum Architecture

Core Body of Knowledge

GSwE2009
Core Materials

Companion Documents

- *Comparisons of GSwE2009 to Current Master's Programs in Software Engineering*

  – Examine real master's programs to determine how well they satisfy GSwE2009 recommendations in order to understand how different GSwE2009 is from current practice

- *Frequently Asked Questions on Implementing GSwE2009*

  – Provide help to faculty who wish to adapt and adopt GSwE2009; e.g., what type of faculty should be recruited; how might a university compensate in the program for students who lack the expected 2 years development experience at program entry

- With continuing support from DoD, Stevens and volunteer authors will maintain these companion documents indefinitely

- With continuing support from DoD and possible support from NSF, Stevens and volunteer authors will actively encourage and enable adoption

17

BKCASE Overview

- Stevens and the Naval Postgraduate School have begun a 3-year project to create a robust body of knowledge and a reference curriculum to advance systems engineering.

- DoD recognizes that their own SE success depends on having a well-accepted robust SE BoK on which standard practice, certification, and workforce competency and education can be based. They are providing substantial funding for effort.

- BKCASE will likely follow similar approach as did SWEBOK and GSwE2009, two analogous projects for software engineering and leverage other efforts such as NPS Modeling and Simulation Acquisition Curriculum

- INCOSE, IEEE Systems Council, IEEE Computer Society Educational Activities Board, and NDIA Systems Engineering Division are participating

BKCASE Authors So Far…

1. Rick Adcock
2. John Baras
3. Barry Boehm
4. Tim Ferris
5. Kevin Forsberg
6. Richard Freeman
7. Sandy Friedenthal
8. Tom Hilburn
9. Scott Jackson
10. Bud Lawson
11. Alex Lee
12. Ray Madachy
13. Ken Nidiffer

14. Dave Olwell
15. Art Pyster
16. Garry Roedler
17. Bill Rouse
18. Jean-Claude Roussel
19. Hillary Sillito
20. John Snoderly
21. Alice Squires
22. Massood Towhidnejad
23. Mary VanLeer
24. Brian Wells

BKCASE Vision and Objectives

Vision

> "Systems Engineering competency models, certification programs, textbooks, graduate programs, and related workforce development initiatives around the world align with BKCASE."

Objectives

1. Create a SEBoK that is globally recognized by the SE community as the authoritative BoK for the SE discipline.

2. Create a graduate reference curriculum for SE (GRCSE – pronounced "Gracie") that is globally recognized by the SE community as the authoritative guidance for graduate programs in SE.

3. Facilitate the global alignment of related workforce development initiatives with SEBoK and GRCSE.

4. Transfer stewardship of SE BoK and GRCSE to INCOSE and the IEEE after BKCASE publishes version 1.0 of those products, including possible integration into their certification, accreditation, and other workforce development and education initiatives.

SEBoK Value Proposition

1. There is no authoritative source that defines and organizes the knowledge of the SE discipline, including its methods, processes, practices, and tools. The resulting knowledge gap creates unnecessary inconsistency and confusion in understanding the role of SE in projects and programs; and in defining SE products and processes. SEBOK will fill that gap, becoming the "go to" SE reference.

2. The process of creating the SEBoK will help to build community consensus on the boundaries and context of SE thinking and to use this to help understand and improve the ability of management, science and engineering disciplines to work together.

3. Having a common way to refer to SE knowledge will facilitate communication among systems engineers and provide a baseline for competency models, certification programs, educational programs, and other workforce development initiatives around the world. Having common ways to identify metadata about SE knowledge will facilitate search and other automated actions on SE knowledge.
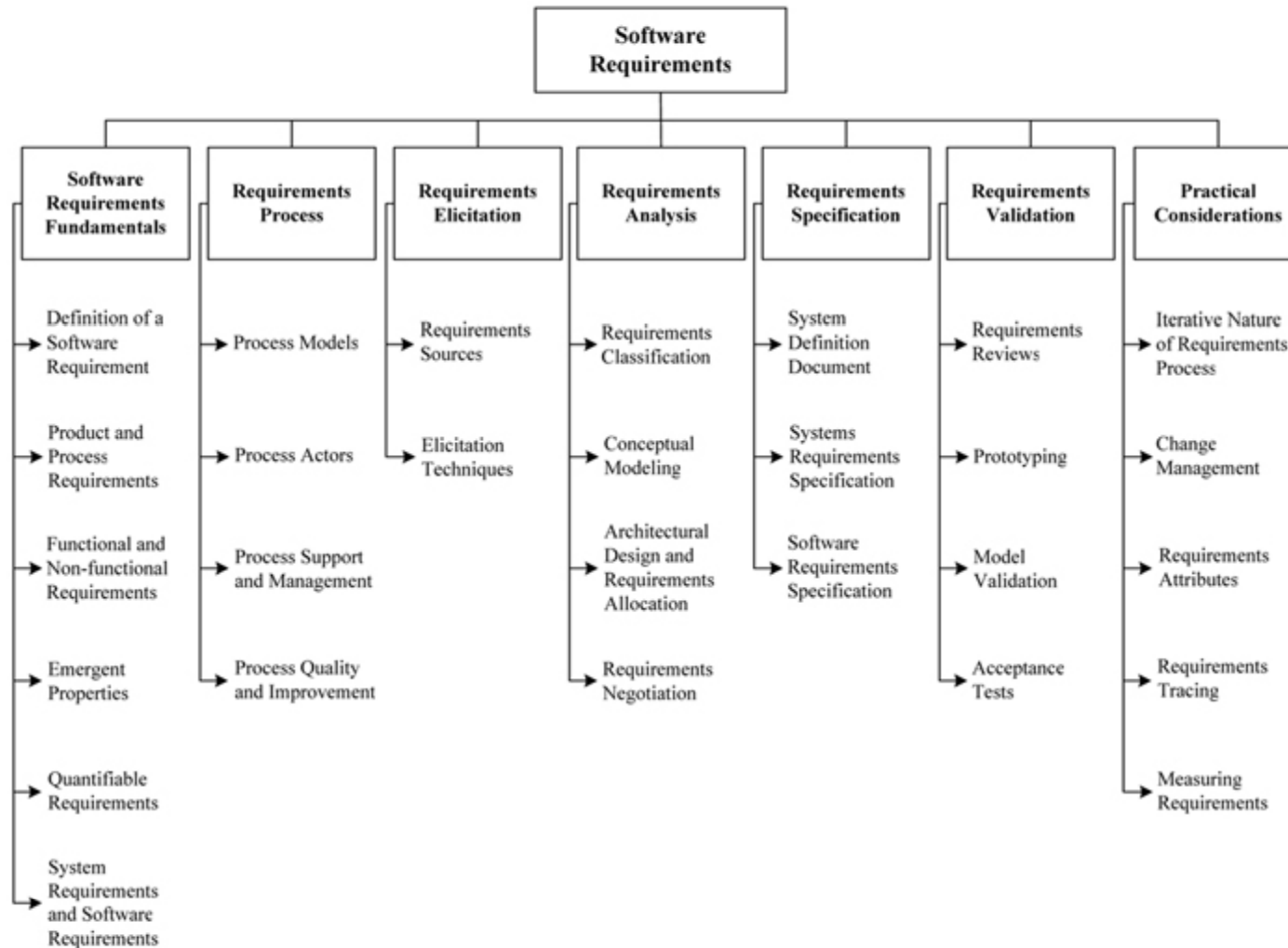
Strategy

1. Publish incrementally/iteratively with GRCSE trailing SEBoK

2. Create common vocabulary to facilitate communications among the team

3. Throughout the project, involve professional societies to facilitate quality, acceptance, and their eventual role as stewards

4. Build early consensus and maintain it throughout the lifetime of the project

5. Rely on and include academia, industry, and government from multiple fields for authors and reviewers

6. Extensively leverage volunteer labor for both authoring and review

7. Rely on existing source material wherever possible and involve principals from efforts that created source material wherever possible

8. Leverage the processes used to create GSwE2009 and the NPS Modeling and Simulation Acquisition Curriculum

9. Keep completely open and collaborative at a global level – but authors make content decisions

10. Hold physical workshops every 3 months to synchronize teams and build team relationships – rely on virtual meetings, email, and other collaboration technology at other times

11. Keep the team focused on the value propositions when conflicts arise.

SWEBOK Example

BREAKDOWN OF TOPICS FOR SOFTWARE

Software Requirements

Software Requirements Fundamentals
- Definition of a Software Requirement
- Product and Process Requirements
- Functional and Non-functional Requirements
- Emergent Properties
- Quantifiable Requirements
- System Requirements and Software Requirements

Requirements Process
- Process Models
- Process Actors
- Process Support and Management
- Process Quality and Improvement

Requirements Elicitation
- Requirements Sources
- Elicitation Techniques

Requirements Analysis
- Requirements Classification
- Conceptual Modeling
- Architectural Design and Requirements Allocation
- Requirements Negotiation

Requirements Specification
- System Definition Document
- Systems Requirements Specification
- Software Requirements Specification

Requirements Validation
- Requirements Reviews
- Prototyping
- Model Validation
- Acceptance Tests

Practical Considerations
- Iterative Nature of Requirements Process
- Change Management
- Requirements Attributes
- Requirements Tracing
- Measuring Requirements

SWEBOK Topic Text

1. Software Requirements Fundamentals

1.1. Definition of a Software Requirement

At its most basic, a software requirement is a property which must be exhibited in order to solve some problem in the real world. The Guide refers to requirements on "software" because it is concerned with problems to be addressed by software. Hence, a software requirement is a property which must be exhibited by software developed or adapted to solve a particular problem. The problem may be to automate part of a task of someone who will use the software, to support the business processes of the organization that has commissioned the software, to correct shortcomings of existing software, to control a device, and many more. The functioning of users, business processes, and devices is typically complex. By extension, therefore, the requirements on particular software are typically a complex combination of requirements from different people at different levels of an organization and from the environment in which the software will operate.

An essential property of all software requirements is that they be verifiable. It may be difficult or costly to verify certain software requirements. For example, verification of the throughput requirement on the call center may necessitate the development of simulation software. Both the software requirements and software quality personnel must ensure that the requirements can be verified within the available resource constraints.

Requirements have other attributes in addition to the behavioral properties that they express. Common examples include a priority rating to enable trade-offs in the face of finite resources and a status value to enable project progress to be monitored. Typically, software requirements are uniquely identified so that they can be over the entire software life cycle. [Kot00; Pfl01; Som05; Tha97]

1.2. Product and Process Requirements ▲

A distinction can be drawn between product parameters and process parameters. Product parameters are requirements on software to be developed (for example, "The software shall verify that a student meets all prerequisites before he or she registers for a course.").

A process parameter is essentially a constraint on the development of the software (for example, "The software shall be written in Ada."). These are sometimes known as process requirements.

Some software requirements generate implicit process requirements. The choice of verification technique is one example. Another might be the use of particularly rigorous analysis techniques (such as formal specification methods) to reduce faults which can lead to inadequate reliability. Process requirements may also be imposed directly by the development organization, their customer, or a third party such as a safety regulator [Kot00; Som97].

Primary Technical Decisions 1-2

1. The SEBoK organizes domain independent SE knowledge. It provides a structure for that knowledge, defines important terms, summarizes important topics, selectively helps users choose among popular alternative methods, facilitates search, printing, and application by its intended users, and identifies references which elaborate more fully on all topics. For Version 0.25, the SEBoK will include a set of primary references based on the expert opinion of the SEBoK authors. For subsequent versions, secondary references may be added.

2. The BKCASE Project will develop recommendations on how INCOSE and the IEEE will maintain and evolve SEBoK in accordance with the BKCASE charter, assuming those organizations become stewards of SEBoK after Version 1.0 is released. Version 1.0 of SEBoK itself will include features to facilitate its maintenance and evolution, including the ability for SEBoK users to readily propose new references and evaluate existing references, as well as readily propose changes to all other aspects of the SEBoK.

Primary Technical Decisions 3-4

3. Primary direct SEBoK users will be (a) practicing systems engineers ranging from novices up through senior experts, (b) those responsible for defining and implementing SE processes within organizations, projects, and programs; (c) those responsible for certifying systems engineers and developing certification programs; (d) customers of SE organizations to help them better select and evaluate those organizations; (e) any project manager, engineer, technologist, researcher, or scientist who needs to know about SE; (f) those who educate and train systems engineers; and (g) the GRCSE author team. The SEBoK will facilitate easy access and use by these different types of users.

4. Secondary SEBoK users will be human resource professionals and other workforce development professionals, senior non-technical managers, and lawyers who will use the SEBoK with the support of systems engineers. The SEBoK will facilitate easy access and use by these users.

5. The ISO/IEC/IEEE 15288 process structure will be the initial architecture for the SEBoK. The authors will divide into several teams. Each team will be assigned non-overlapping subsets of 15288 processes. Each team will independently develop initial SEBoK content for their process subset, including methods, techniques, and primary references, taking into account primary and secondary SEBoK users. At Workshop 2, the results of the individual team efforts will be jointly evaluated by the entire author team leading to a revised architecture.

6. Version 0.25 of the SEBoK will be domain independent. Domain dependent knowledge will be captured through case studies of individual systems within specific domains. Those case studies will be companion documents to Version 0.25. After Version 0.25 is complete, the decision to use case studies as the only means to capture domain specific knowledge will be revisited.

GRCSE Value Proposition

- There is no authoritative source to guide universities in establishing the outcomes graduating students should achieve with a master's degree in SE, nor a guidance source on reasonable entrance expectations, curriculum architecture, or curriculum content.

- This gap in guidance creates unnecessary inconsistency in student proficiency at graduation, makes it harder for students to select where to attend, and makes it harder for employers to evaluate prospective new graduates.

- GRCSE will fill that gap, becoming the "go to" reference to develop, modify, and evaluate graduate programs in SE. <do we want to include domain specific SE programs as well as SE discipline?>

INCOSE Curriculum Framework

"A Reference Curriculum for a Graduate Program in Systems Engineering" by Jain, Squires, Verma, and Chandrasekaran, 2007

Specialization Courses
- Software Systems Engineering
- General Project Management
- Finance, Economics, and Cost Estimation
- Manufacturing, Production, and Operations
- Organizational Leadership
- Engineering Ethics and Legal Considerations
- Masters Project or Seminar

- Systems Design/Architecture
- Systems Integration and Test
- Quality, Safety, and Systems Suitability
- Modeling, Simulation and Optimization
- Decisions, Risks and Uncertainty

- Fundamentals of Systems Engineering
- Intro to Systems Engineering Management

- General Mathematics
- Probability & Statistics

Level 3: (9 Credits)

Level 2: Core Courses (12 Credits)

Level 1: Introductory Courses (9 Credits)

Level 0: Foundation Courses

What is Systems Engineering?

- Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems. (INCOSE)

- Systems Engineering integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. It considers both the business and technical needs of all customers with the goal of providing a quality product that meets the user needs. (INCOSE)

- Systems engineering is a robust approach to the design, creation, and operation of systems. In simple terms, the approach consists of identification and quantification of system goals, creation of alternative system design concepts, performance of design trades, selection and implementation of the best design, verification that the design is properly built and integrated, and post-implementation assessment of how well the system meets (or met) the goals. (NASA)